The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 258 - 266

SVM-Based Machine Learning Approach for Mobile Botnet Detection

Smita S. Wagh
Department of Computer Engineering, JSPM'S Jayawantrao Sawant College of Engineering, Pune, India.
smitawagh191992@gmail.com
Mohini A. Bhokare
Department of Computer Engineering, JSPM'S Jayawantrao Sawant College of Engineering, Pune, India.
mbhokare2002@gmail.com
Fazila F. Khan
Dept. of Computer Engineering, JSPM'S Jayawantrao Sawant College of Engineering, Pune, India.
k.fazila66@gmail.com
Rahul G. Nawale
Dept. of Computer Engineering JSPM'S Jayawantrao Sawant College of Engineering, Pune, India.
rgnawale10@gmail.com
Maaz I.Quadri,
Dept. of Computer Engineering, JSPM'S Jayawantrao Sawant College of Engineering, Pune, India
maazquadri530@gmail.com

Abstract: Botnets pose a significant threat to global Internet security. A new pattern is evolving to shift botnets from conventional desktop to mobile platforms with continued sophistication and durability surroundings that move. To lessen the harm that mobile botnets pose, just like in the desktop environment, detection is crucial. Recognizing patterns of unusual behaviour is one of the many methods used to find these botnets, and it produces the best and most often findings. Analysing the running characteristics of this kind of application is one technique to spot similar tendencies in the mobile botnet area. This article examines host-based and anomaly-based methods for detecting mobile botnets. The suggested method extracts statistical characteristics of aberrant behaviour from system calls and utilises machine learning methods to identify them. In realistic settings, the effectiveness of the employed strategy can be tested. Excellent outcomes were attained by the suggested strategy, which included a low false positive rate and a high actual detection rate.
Keywords: Machine Learning, Support Vector Machine (SVM), Mobile Botnet, Botnet Detection, Deep Learning, Security, Classification.

I. Introduction

The technology that is currently most prevalent in our daily lives and is used by many people is Android. Every human owns and often employs an android. Modern mobile devices are seamlessly integrated with cutting-edge knowledge and technologies, like the Internet. Mobile security is a crucial concern globally due to the widespread use, ease, and mobility of mobiledevices today[12].

In recent years, both personal and professional use of smartphones has significantly

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 258 - 266

increased1. Global shipments of consumer smartphones reached 349 million units in the first quarter of 2016, up 3.9% from the same period in 2015, according to Gartner Inc.2 [1]. Botnet detection commonly employs two techniques: anomaly-based and signature-based. For detection, signature-based techniques rely on the particular signatures of malware and botnets [2].

Naive Bayes (NB), Support Vector Machines (SVM), Random Forest (RF), J48 Decision Tree, and Simple Logistic are among the classifiers examined (SL). 350 features from a static analysis of each app in the corpus[3] were used to train these systems. CNNs can be used with one-dimensional data, but they are more frequently used with multidimensional data, which makes them successful for image and video analysis based problems [4].

Therefore, a collection of compromised smartphones and other mobile devices that are remotely managed by botmasters via CC channels is referred to as a "mobile botnet". Here is a machine learning programme for this issue that utilises SVM to find Android botnets. The SVM model uses static characteristics that are dependent on a specific data set to categorise new or previously undiscovered apps as either discovered apps.

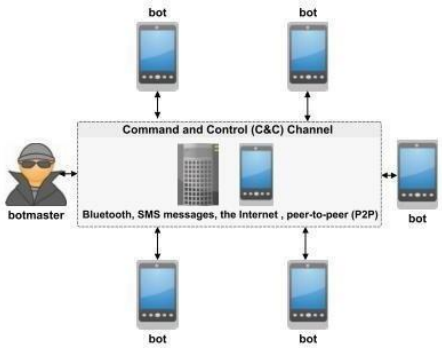

Figure1. Mobile Botnet Architecture[2]

II. Literature Survey

People are using smartphones more frequently as a result of their increasing importance in daily life. However, this also encourages hackers to create harmful software, particularly Android botnet, to steal private information and cause financial harm. because the technologies employed by developers of maliciousapplications (apps) change quickly[1].

1. Datasets: The training dataset and testing dataset were both employed in this study. A detection model is built using a training dataset, and the model is validated using a testing dataset.
2. Reverse Engineering Process: The key tasks during this stage are decompressing and decompiling the APK files. The term "APK" stands for "Android Application Package," which refers to a compressed (zipped) file ofany Android application.
3. Feature Extraction: This pre-processing module is extracting features for which 2355 Android applications have asked for authorization (botnet and benign).
4. Permission Vector: For each of the 2355 Android applications, the outcome of the feature extraction phase is transformed into a vector in this phase.
5. Permission Selection: The Android operating system has 138 permissions in total17, but not every one of them is requested by the Android applications in the dataset. The subsequent collection of these many permissions may increase storage requirements and processing complexity.

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 258 - 266

6. Classification: J48, Random Forest, and Naive Bayes algorithms are employed. WEKA is used to conduct thepermission selection and classification phases of the data analysis module.

As a result, Android botnet apps pose a severe threat to smartphones because they are more dangerous than other Android malware. The research-proposed method employs the Android app's permissions as features to distinguish between botnet apps and good apps. The most important permissions are chosen using the Information Gain technique, and the Android apps are then classified as botnet or benign using the Nave Bayes, Random Forest, and J48 classification algorithms. The testing results reveal that the Random Forest algorithm had the lowest false positive rate (0.099) and the highest detection accuracy (0.946, or 94.6%).

One of the most well-known and widely used operating systems for smartphones is Android. Numerous millions of its applications are available at both official and unofficial shops. These stores can be used to publish botnet programmes, a type of malware that victims can download to their smartphones[2].

1. Context analysis: Understanding the issue with mobile botnets and identifying their components is the goal of this step.
2. Data Collection: The information needed to train and test our machine learning models is gathered from two main sources: Android Botnet applications and Android Benign applications. Information regarding Android botnet apps was gathered using a dataset of 1635 botnet applications from the ISCX Android Botnet, which consists of 14 different botnet families.
3. Feature Extraction: Using the Android requested permissions and its suitable protection level categories 4, we extracted 145 features to be used in training the classifiers.
4. Model development: In this stage, we build our detection models using four well-known machine learning classifiers: Random Forest (RF), Multilayer Perceptron neural networks (MLP), Decision trees (J48), and Nave Bayes (NB) classifiers. All models are built on the basis of two datasets. The second has the capability related to permissions protection levels, but the first does not.
5. Evaluation and assessment: The three assessment metrics utilised to evaluate the effectiveness of the created models are accuracy rate, precision, and recall. These metrics may be computed using the confusion matrix.

Result: In this work, we examined the problem of identifying Android botnets using static analysis and machine learning. Additionally, we developed all required tools for crawling the Google Play store, downloading APK files, scanning the APK files to generate a dataset of benign applications, disassembling the APK files, and developing a tool for extracting 145 features for all requested permissions features available in Android as well as a brand-new set of features for the permissions protection levels from the decompressed APK folders. We evaluated the performance for two sets of attributes using Random Forest, Multilayer Perceptron neural networks, Decision trees, and Naive Bayes classifiers. The experimental results show that the Random Forestclassifier delivers good results for both feature groups.

More and more malware is concentrating on the most widely used mobile operating systems. Malicious software that turns smartphones and other mobile devices into botnet-capable machines raises serious security concerns. This calls for the creation of more trustworthy methods for Android botnet identification. Therefore, in this paper, we provide a deep learning approach for Android bot-net identification based on convolutional neural networks (CNN). Our suggested botnet detection solution is built as a CNN-based model that is trained on 342 static app parameters to distinguish between botnet pps and ordinary apps[3].

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 258 - 266

1. Dataset: This work made use of the Android dataset from [5], also known as the ISCX botnet dataset. In past studies like [4], [7-10], and, the ISCX dataset—which comprises 1,929 botnet apps from 14 families—was utilised. A total of 4,873 clean applications were used in this study and classified as "normal" to facilitate supervised learning when training the CNN and other machine learning classifiers. On the Google Play store, the clean applications were downloaded from various app categories, and Virus Total was used to verify their safety. The 5 distinct categories of the 342 static characteristics for model training that were extracted from the apps: : API calls, commands, permissions, intents, and extras are just a few examples.
2. Tests of the proposed CNN-based model's evaluation We conducted many sets of tests to evaluate the effectiveness of our proposed model. We used the measures accuracy, precision, recall, and F1-score to evaluate model performance. The following metrics are established (with the botnet class being deemed favourable): • Accuracy: This concept is the ratio of successfully anticipated outcomes to all other outcomes. • Accuracy: The sum of all actual good things and all anticipated good things. Was the model's pessimistic prognosis, for instance, correct? Recall is calculated by dividing true positives by all real positives.

Result: In this work, we proposed a deep learning model based on 1D CNN for the aim of recognising Android botnets. We thoroughly tested the model with 1,929 botnet applications and 4,387 clean apps. The model outperforms a variety of well-known machine learning classifiers on the same dataset. Our proposed CNNbased model outperforms the other models in terms of accuracy, precision, recall, and F1-score, demonstrating that it may be used to identify new, previously undiscovered Android botnets.

III. Proposed System

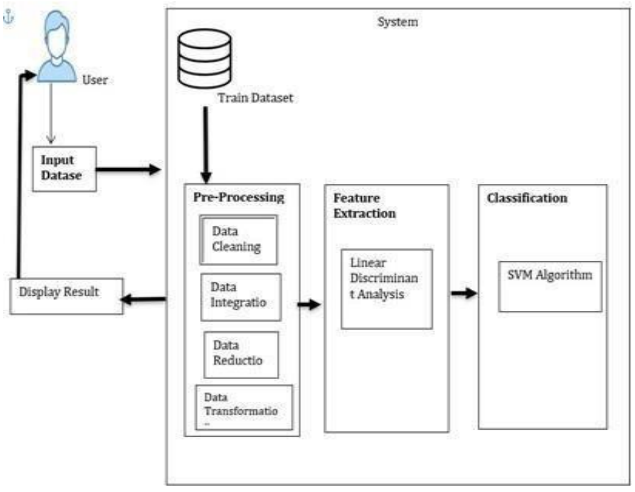

Figure 2: Architecture of proposed System

Cortes and Vapnik created the SVM [7]. It is a supervised learning model based on Vapnik Chervonenkis dimension and structural risk minimization. SVMs are frequently employed to tackle classification or regression issues in machine learning. As a result, the primary goal of SVM is to choose the best hyperplane for examining different categorization data. As demonstrated, the biggest fringes connected to various classification data are found in the ideal hyperplane. The two forms of categorization data are represented by 2 black points and 3 white points on the maximal boundary; these points are referred to as support vectors. SVM

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 258 - 266

selects vectors or extrema that infact form a hyperplane.

The methods in this approach are referred to as support vector machines because of these extreme situations, which are known as support vectors. The CNN method has been employed in several prior research for comparable objectives, and in this work we leverage its SVM to raise accuracy and enhance consistency of results. This is crucial. The example below can be used to teach SVM. Let's say you come across an unusual, strange cat that resembles both a dog and a cat in some ways. The SVM technique can therefore be used to create a model that accurately determines whether a specific animal is a cat or a dog. To understand the many traits of cats and dogs, we first train a model using a vast number of cat and dog photos. The SVM then generates the alien creature for the test using the same design as before. These two dates are where the decision border falls.

The three different kinds of kernel functions are sigmoid, polynomials, and radial basis functions (RBFs). To expedite the categorization, the data must be transformed using a suitable kernel function. SVM algorithms can be used for tasks including text classification, face detection, and picture classification. Administrators are required to enter a valid login and password to log in. The user can upload recordings to the system once they have logged in successfully. The specified dataset is then processed using the training dataset. All pre-processing tasks, including data integration, data reduction, and data transformation, are completed and moved on to the following stage.

Extraction of Features: Features are extracted.

Classification: The SVM algorithm does classification. If the problem record is a "botnet app" or a "normal app," the SVM emits output.

Steps to follow :

Step 1: The system must verify the user's login or registration information.

Step 2: Keep the record in your system.

Step 3: Feature extraction from system-pre- processed data.

Step 4: The support vector machine technique isthen used for classification.

Display the outcome "botnet" or "not botnet" in step five.

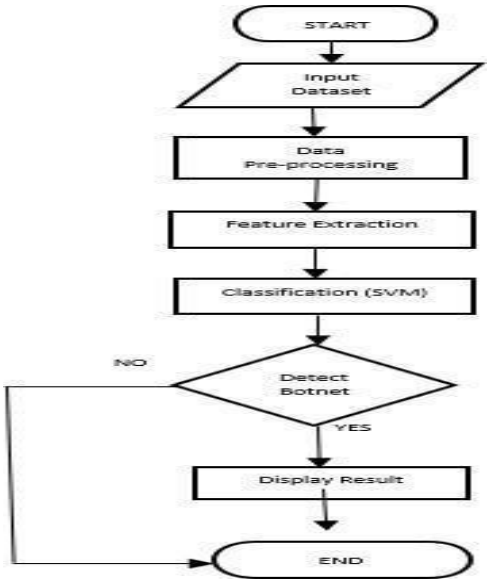

Figure 3: System Flow Chart

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 258 - 266

To address classification and regression problems, SVM (Support Vector Machine) is a supervised machine learning technique. But it's routinely used to deal with categorization problems. Each piece of data is represented as a point in an n-dimensional space, where n is the number of features, and the value of each feature correlates to a certain position in the SVM method. Finding a hyperplane that clearly demarcates the two classes is the first step in classification.

Classification and regression problems may be solved using a supervised machine learning technique called SVM (Support Vector Machine). It is, nevertheless, widely used to overcome categorization problems. Each piece of data is represented as a point in an n dimensional space (n being the number of features), where each feature's value is mapped to a specific coordinate value using the SVM method. Finding a hyperplane that distinctly divides the two classes is how classification is done.

B) There are two primary SVM subtypes:

1. Linear SVM - As the name suggests, linear SVM uses all linearly separable data. H. When a dataset can be split into two classes by a straight line, it is said to be linearly separable, and the corresponding classifier is known as a linear SVM classifier.
2. Nonlinear SVM: As its name suggests, nonlinear SVM is likewise highly helpful for data that has been divided into categories using linear separation. In other words, if a dataset cannot be categorised using a straight line and a nonlinear SVM classifier is used to do it, the dataset is said to be nonlinear.

C) SVM as opposed to CNN: (Support Vector Machine)

1. Any collection of data may have certaincharacteristics removed using SVM.
2. In some situations, SVM may also choosecertain dataset attributes.
3. SVM can easily handle small datasets andprevent problems like overfitting.
4. The binary classification precision of the SVMmethod is 80.95%.
5. The accuracy of SVM multiclass classificationis about 50%.

SVM ALGORITHM

Step 1: Load the important libraries

Step 2: Import dataset and extract the X variables and YSeparately.

Step 3: Divide the dataset into train and test Step 4: Initializing the SVM classifier modelStep 5: Fitting the SVM classifier model Step 6: Coming up with predictions

Step 7: Evaluating model’s performance various kernel functions[10]

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 258 - 266

PERFORMANCE ANALYSIS

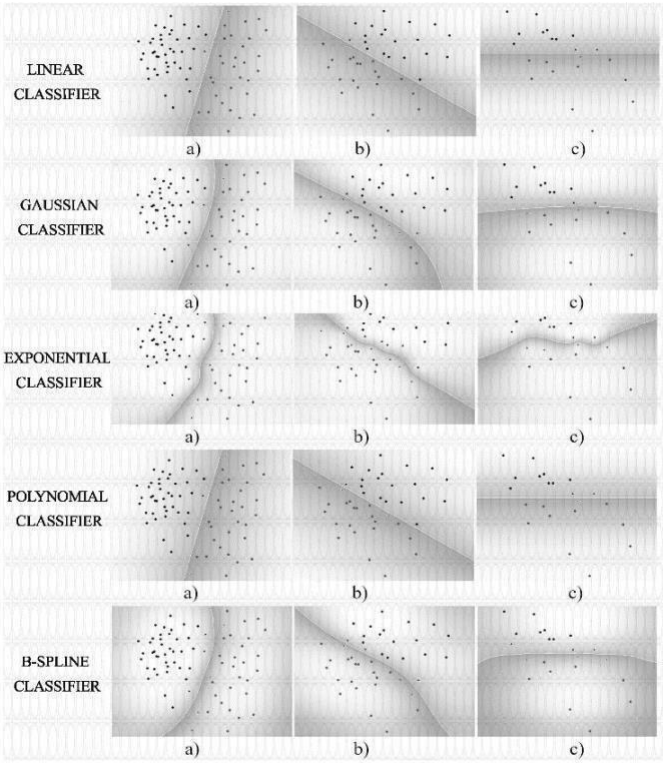

Figure 4: SVM classification results with

The identified objects are split into two categories in the initial iteration: malignant conduct and benign activity. After that, the classifier further divides the objects into two classes. For instance, spambots and bad behaviour. Isolate malicious behaviour and other classes from malware, etc., in the following cycle, untileverything is entirely separated.

Tool Used : An open source tool Android Device monitor (ADM) was used to analyse the performance of system. It provides a graphical user interface for several android application debugging and analysis tools.

Classifier kernel	Classification accuracy		Execution time (sec)	Distance between hyperplanes
	one against	one against all SVM		
linear	92.53	96.35	0.5	0.18
Gaussian	96.18	97.39	0.4	0.25
B-spline	98.01	97.43	0.4	0.38
polynomial	93.56	96.79	0.5	0.37
exponential	97.02	97.64	0.5	0.33

Table 1 : displays the experimental findingsfor various SVM classifiers[10].

EXPECTED RESULT

Test results for several mobile malware classes: Sensitivity (SN), specificity (SP), training set sample size T, evaluation sample size E (containing malicious and benign traffic samples), and suggested technique Reconstruction success rate, true positives (TP), true negatives (TN), false positives (FP), and overall accuracy (Q) (SR).

The Ciência & Engenharia - Science & Engineering Journal
ISSN: 0103-944X
Volume 11 Issue 1, 2023
pp: 258 - 266

Mobile malware's classes	T	E				Results		
		Malicious		Benign		SN, %	SP, %	Q, %
		TP	FN	TN	FP			
trojans	36	38	3	34	1	92,68	97,14	94,74
backdoors	31	33	2	29	2	94,29	93,55	93,94
mobile botnets	28	27	1	25	0	96,43	100,00	98,11
spammers	32	34	3	35	2	91,89	94,59	93,24
spyware	26	29	2	30	0	93,55	100,00	96,72
smartphone trackers	37	39	3	35	3	92,86	92,11	92,50
mobile proxy-servers	24	30	0	25	1	100,00	96,15	98,21
SMS malware	41	40	3	39	2	93,02	95,12	94,05
exploits	35	36	3	34	2	92,31	94,44	93,33
rootkits	30	35	6	30	1	85,37	96,77	90,28
adware	34	27	2	28	2	93,10	93,33	93,22
DDoS attackers	29	35	2	30	1	94,59	96,77	95,59

Table 2 : The overall accuracy of technique[10]

Above table presents overall results of techniques efficiency taking into account sensitively specificity and the overall accuracy. Employing the dataset this stage involved the evaluation of the SVM based inference engine accuracy concerning each type of malware separately. The combine result are given in table 2.

Table 2 describes the overall accuracy of technique is in the range from 90.28% to 98.21%. Moreover, sensitively specificity are in the range from 85.37 – 100% and 92.11 – 100% respectively. Therefore this approach indicates the capability of SVM for the botnets classification.

IV. Conclusion & Future Work

Malware threats in society include botnets. They are employed to steal information, compromise systems, and harm and destroy systems. They are challenging to find and get rid of. Therefore, our method is a helpful tool for identifying mobile botnets.

Machine that supports vectors In order to accurately detect and improve the accuracy of the prior system, algorithms have been utilised. This model increases accuracy from the prior system's 85% to 96.77%.

More data sets can be uploaded here to make it even more accurate and user-friendly in the event that malicious applications are subsequently detected. Even while it cannot hold or reach that level, it can be used in higher-level platforms like the Play Store and AppStore because to these capabilities. To make sure that all applications and programmes are malware-free, it can also be employed in local company systems.

References

[1] Zubaile Abdullah, Madihah Mohd Saudi, Nor Badrul Anuar,"ABC : Android Botnet classification using Feature Selection Classification Algorithms" May 2017 Journal of Computational and Theoretical Nanoscience4717-4720

[2] Wadi ijawi .Ja'far Alqatawna Hossam Faris "Toward a Detection Framework for Android Botnet" October 2017.

[3] Y. Yerima and S. Khan "Longitudinal Performance Analysis of Machine Learning based Android Malware Detectors" 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), IEEE

[4] Dhabliya, D., & Dhabliya, R. (2019). Key characteristics and components of cloud computing. International Journal of Control and Automation, 12(6 Special Issue), 12-18.

[5] Suleiman Y.Yerima1 and Mohammed K. Alzaylaee2"Mobile Botnet Detection: A Deep Learning Approach Using Con- volutional Neural Networks".1 June 2020,IEEE

[6] Kadir, A.F.A., Stakhanova, N., Ghorbani, A.A., 2015. Android botnets: Whaturls are

telling us, in: International Conference on Network and System Security, Springer. pp. 78–91.

[7] M. Eslahi, R. Salleh, and N. B. Anuar, ”Bots and botnets: An overview of characteristics, detection and challenges,” in Proceedings of the IEEE International Conference on ControlSystem, Computing and Engineering(ICCSCE), 2012, pp. 349-354.

[8] J. Dae-il, C. Kang-yu, K. Minsoo, J. Hyun- chul, and N. Bong-Nam, “Evasiontechnique and detection of malicious botnet,” in Proceedings of the Internet Technology and Secured Transactions (ICITST), 2010 International Confer- ence for, 2010, pp. 1-5

[9] Z. Yajin and J. Xuxian, “Dissecting Android Malware: Characterization and Evolution,” in Proceedings of the Symposium on Securityand Privacy (SP), 2012, pp. 95-109.

[10] Dhabliya, D., & Parvez, A. (2019). Protocol and its benefits for secure shell. International Journal of Control and Automation, 12(6 Special Issue), 19-23.

[11] M. Eslahi, M. Rohmad, H. Nilsaz, M. Var Naseri, N. Tahir, and H.Hashim, ”Periodicity classification of HTTP traffic to detect HTTP Botnets,” in Pro- ceedings of the Computer Applications Industrial Electronics (ISCAIE), 2015 IEEE Symposium on, 2015, pp. 119-123.

[12] T. Strazzere and T. Wyatt, “Geinimi trojan technical teardown,” Lookout Mo-bile Security, 2011.

[13] Gu, R. Perdisci, J. Zhang, and W. Lee, “BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection,” inProceedings of the 17th conference on Security symposium, San Jose, CA, 2008,pp. 139-154.

[14] M. Eslahi, R. Salleh, and N. B. Anuar, ”MoBots: A new generation of botnet on mobile devices and networks,” in Proceedings of the IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE), 2012, pp. 262- 266